

Chapter 24: UPD/UPP Command Reference

This chapter contains full usage information on all the **UPD** commands and the **UPP** command. In particular, for each command you will find:

- a statement of the purpose and/or function of the command
- the command syntax
- a listing of commonly used options, without descriptions
- a listing of all valid options, with command-specific descriptions
- (as needed) a section called “Options Valid with -G”
- (as needed) a section called “More Detailed Description” which typically includes detailed command-specific usage information
- command examples

For commands that have a corresponding **UPS** command, you will find:

- a statement of the purpose and/or function of the command
- the command syntax
- a reference to the corresponding **UPS** command
- a listing of any additional, **UPD**-specific options
- (as needed) command examples

For the **upd addproduct** and **upd install** commands we include a detailed list of the internal processes the command performs. The internal processes for the other commands can largely be inferred from these lists.

24.1 upd addproduct

The **upd addproduct** command adds a product instance to a product distribution database. A product instance may contain any or all of the following: a product root directory, a table file, and/or a **ups** directory. The product may be in tar file format or unwound. **upd addproduct** declares the product instance on the distribution node with the same product instance identifiers (e.g., product name, version, flavor, qualifiers, chain) as it has on the local node.

24.1.1 Command Syntax

For adding a product that is declared to a local database

An unwound product or a table file:

```
% upd addproduct [<flavor_option>] [<other_options>] <product> \
  \ <version>
```

A product tar file:

```
% upd addproduct [<flavor_option>] -T <archFilePath> \
  [<other_options>] <product> <version>
```

Note: The flavor option is not strictly required; **upd addproduct** defaults to the current instance on the local system, similarly to other **UPS** and **UPD** commands.

For adding a product that is not declared to a local database

An unwound product:

```
% upd addproduct [-P] <flavor_option> -r <prodRootDir> \
  -m <tableFileName> [-M <tableFileDir>] [-U <upsDir>] \
  [<other_options>] \ <product> <version>
```

A table file:

```
% upd addproduct [-P] <flavor_option> -m <tableFileName> \
  [-M <tableFileDir>] [<other_options>] <product> <version>
```

A product tar file:

```
% upd addproduct [-P] <flavor_option> -T <archFilePath> \
  -m <tableFileName> [-M <tableFileDir>] [<other_options>] \
  <product> <version>
```

Notes:

- The **-P** option is not strictly required, however it is recommended. In cases where the local database contains a product instance that also matches the specified options, **-P** ensures that **UPD** ignores the database and picks up the intended instance.
- If the product includes a table file, you must include the **-m** option specifying its name, as there is no default. You must also include **-M** if the table file is not in the current directory.
- If the `ups` directory is in the default location (`${PRODUCT}_DIR/ups`), **UPS** should be able to find it, but it's safer to always specify it via the **-U** option.

24.1.2 Commonly Used Options

See section 24.1.3 *All Valid Options* for descriptions of each option.

For adding a product that is declared to a local database

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (alone or together with one of **-0**, **-1**, **-2**, **-3**, **-4**)
-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**
-h <host>
-q <qualifierList>
-T <archFilePath>
-z <databaseList>

For adding a product that is not declared to a local database

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** together with one of **-0**, **-1**, **-2**, **-3**, **-4**
-h <host>
-m <tableFileName>
-M <tableFileDir>
-P
-q <qualifierList>
-r <prodRootDir>
-T <archFilePath>
-z <databaseList>

24.1.3 All Valid Options

-? (" -?" for **csh)** Prints command description and option usage information to screen
-A <nodeList> Specifies one or more nodes on which use of product is authorized (passed through to **UPS**)
-c Finds product instance chained to “current” on local node, and chains it to “current” on the distribution node. If **-P** used, ignores any local chain, chains selected instance to “current” on the distribution node.

- d** Finds product instance chained to “development” on local node, and chains it to “development” on the distribution node. If **-P** used, ignores any local chain, chains selected instance to “development” on the distribution node.
- D "<origin>"** Specifies the product's master source file for informational purposes. Not fully implemented; currently just passed through to **UPS**.
- f <flavorList>** Described below under “The flavor options”.
- g <chainName>** Finds product instance chained to **<chainName>** on local node, and chains it to **<chainName>** on the distribution node. Can be a list of chain names. If **-P** used, ignores any local chain, chains selected instance to “**<chainName>**” on the distribution node.
- h <host>** Specifies product distribution host to which you are adding the product; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
 - a plain host name; we recommend using the full host name with **upd addproduct -h <host>** (e.g., *fred.fnal.gov*, rather than just *fred*) to prevent problems when people download the product to off-site user nodes. (Using *fred* by itself in **upd addproduct** is possible only if the *sub.domain* is *fnal.gov*; in other commands it is fine to use it that way.)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi`)
- H <flavorList>** Described below under “The flavor options”.
- i** Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one, operation continues to the next.
- m <tableFileName>** Specifies table file name on local node. Required for products that are not declared locally (or when **-P** used); if product is declared locally, can be used to override the corresponding value set in the local declaration.
- M <tableFileDir>** Specifies table file directory on local node.

Generally used with products that are not declared locally (or when **-P** used). In this case, it is required whenever the table file is in a directory other than the current directory.

- If product is declared locally, can be used to override the corresponding value set in the local declaration.
- n** Finds product instance chained to “new” on local node, and chains it to “new” on the distribution node. If **-P** used, ignores any local chain, chains selected instance to “new” on the distribution node.
 - o** Finds product instance chained to “old” on local node, and chains it to “old” on the distribution node. If **-P** used, ignores any local chain, and chains selected instance to “old” on the distribution node.
 - O "<flags>"** Sets the value of \$UPS_OPTIONS to **<flags>**. This value would get passed to the host’s updconfig file. This is not used on *fnkits* (see section 24.1.5 *Adding Products to fnkits.fnal.gov*).
 - p "<description>"** Specifies product description to set in declaration on distribution node. Not fully implemented; currently just passed through to **UPS**.
 - P** Prevents **UPS** from searching in a local database, and thus requires **UPS** to rely solely on information supplied on the command line to locate the product instance to upload. Even if selected instance happens to be declared locally, all database information is ignored.
 - q <qualifierList>** Finds product instance on local node with the specified qualifiers (required and/or optional), and sets instance’s qualifiers on distribution node.
 - r <prodRootDir>** Specifies the local product root directory (generally used with unwound products that are not declared locally; if product is declared locally, can be used to override the product root directory set in the local declaration)
 - s** Lists what command would do; but does not execute the command
 - t** Finds product instance chained to “test” on local node, and chains it to “test” on the distribution node. If **-P** used, ignores any local chain, chains selected instance to “test” on the distribution node.
 - T <archFilePath>** Specifies location of archive file on local node
 - U <upsDir>** Specifies location of **ups** directory on local node; default value is **ups**, relative to the product root directory (generally used with products that are not

declared locally; if product instance is declared locally, can be used to override the `ups` directory path set in the local declaration)

- v(vvv)** Prints out extra debugging information.
- V** Does not delete the temporary script files or partially installed products when command finishes; instead lists them on the screen
- z <databaseList>** Specifies the local database(s) in which to look for the product instance to upload

The flavor options

Flavor may be specified using **-f**, using **-H** by itself or in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).

- f <flavorList>** Finds product instance of specified flavor on local node, and sets instance's flavor on distribution node. If multiple values specified, **upd addproduct** is run once for each flavor.

- H <flavorList>** Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families. If multiple values specified, **upd addproduct** is run once for each value.

Can be used alone (without an accompanying number option) if product instance is locally declared and **-P** is not used. In this case, the best match instance of the flavor family on the local node is picked, and it gets declared on distribution node as that same "best match" flavor.

If used with any of **-0**, **-1**, **-2**, **-3**, **-4**, product instance gets declared on distribution node as specified level of that flavor; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd addproduct** is run once for each flavor, according to the accompanying number option.

- 0** Specifies flavor as NULL; equivalent to **-f NULL**
- 1** Specifies flavor as OS value up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.

- 2 Specifies flavor for product instance on local and distribution nodes as OS value up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.
- 3 Specifies flavor for product instance on local and distribution nodes upto the release of the version; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4 Specifies flavor for product instance on local and distribution nodes upto the patch number of the release; e.g., equivalent to **-f SunOS+5.6.2**; if given together with **-H IRIX+6.2.1**, flavor is then specified as IRIX+6.2.1.

24.1.4 More Detailed Description

About the tar file

It is optional to create a tar file of your product prior to running **upd addproduct**. **upd addproduct** will create one for you if it knows the location of the product root directory. It can find this information in two ways:

- If the product instance has been declared to a local **UPS** database listed in **\$PRODUCTS** ahead of time, and the product root directory (**PROD_DIR**) appears in the declaration, **UPD** can pick it up. You can check this using **ups list -l**.
- You can supply the product root directory on the **upd addproduct** command line using the **-r** option.

When it is left to **UPD** to create the tar file, it makes the tar file on the local node in the local **\$TMPDIR** area. In the current release of **UPD**, you cannot choose which files to include in a tar file made this way; all files get included except **CVS** directories and core files. The old-style **upd_files.dat** is obsolete.

upd addproduct unwinds the **ups** directory on the distribution node, if it was included in the tar file, thereby making the directory and its contents available for individual file retrieval via **upd fetch** (see section 24.6 *upd fetch*).

About Chains

Chain information remains identical for the added product instance on the local and distribution nodes under most circumstances. If **-P** is used, local chain information is ignored, but can be set on the distribution node. You can use **upd modproduct** afterwards to change the chain (see section 24.10 *upd modproduct*).

Internal Processes

The **upd addproduct** command operates by making a series of network connections to the server. All calls are made from the client system to the distribution server, who reports back results on the same data channel:

- The Web server on the distribution node is called, and a script called `ups.cgi` is used to determine if the specified product instance already exists on the distribution node. If it exists, **UPD** on the client machine prints an error and exits. If it doesn't exist, the process continues.
- The anonymous **FTP** server on the distribution node is called, and the product tar file (if any) is transferred from the user node into `/incoming`.
- The Web server is called, and `upd.cgi` is used to call `upd move_archive_file`. This script makes a product directory for the instance on the distribution node (as defined by the distribution node's `updconfig` file), installs the tar file as `${UPS_PROD_DIR}.tar` (or `${UPS_PROD_DIR}.tar.gz` or `${UPS_PROD_DIR}.zip`, according to its suffix), and unwinds part of the tar file (to make the `README` file and the `ups` and `man` directories available, if present).
- The script `upd.cgi` reports back the database, product directory, and tar file location to the client **upd addproduct** command.
- The anonymous **FTP** server is called, and the product `ups` directory tar file is uploaded to `/incoming`. (If the user specified a `ups` directory, it gets uploaded over the one that was unwound from the tar file.)
- The Web server is called, and `upd.cgi` is used to call `upd moved_ups_dir`. This script makes a `ups` directory on the distribution node for the product (as defined by the `updconfig` file) and unwinds the `ups` directory tar file.
- The script `upd.cgi` reports back the database and `ups` directory to the client **upd addproduct** command.
- The **FTP** server and Web server are similarly called to install the table file.
- Finally, the Web server is called and `ups-decl.cgi` is used to declare the product into the distribution database.

A subset of these steps is performed to execute **upd modproduct** or to add a product that has a subset of these elements (e.g., one that does not include a `ups` directory).

24.1.5 Adding Products to **fnkits.fnal.gov**

The central Fermilab Computing Division product distribution server, *fnkits.fnal.gov*, recognizes several different categories of product:

default	regular products added to the <code>KITS</code> database for distribution to any on-site or registered off-site node ¹ .
FermiTools	locally-developed and supported software packages that we make available to the public
proprietary	products for which Fermilab has a limited number of licenses

1. See the **Product Distribution Platform Registration Request** form at http://www.fnal.gov/cd/forms/upd_registration.html.

<code>fnalonly</code>	products accessible only to the <code>fnal.gov</code> domain
<code>usonly</code>	US-only (United States only) products are accessible only to U.S. government (<code>.gov</code>) and military (<code>.mil</code>) domains

Most products fall into the default category, and can be added normally. For the other categories, you must first fill out the **Special UPD Product Registration** form (at <http://fnkits.fnal.gov/specialprod.html>) indicating which category of product it is, and submit the form. Then when you receive an email message saying that your product has been registered as a special product, go ahead and add it to *fnkits*. Do not use any special options (i.e., `-O "options"`) with `upd addproduct`; your product will automatically be configured to handle the special requirements according to your selection on the form.

24.1.6 upd addproduct Examples

Add locally-declared product using defaults

```
% upd addproduct foo v1_0 -2
```

UPD looks in `$PRODUCTS` to find the product `foo v1_0` for the `-2` flavor level of the local machine. If necessary, it makes a tar file, adds it to the default distribution node *fnkits*, and declares it there. This will work if:

- the local database in which `foo` is declared is listed in `$PRODUCTS` (necessary for tar file creation)
- `foo` has its `ups` directory (in addition to all the product files) under the product root directory, and
- its table file is in one of the default locations (under either the `ups` directory or `$PRODUCTS/foo`).

If the command succeeds, **UPD** returns a message indicating that the product was successfully transferred and declared.

Add locally-declared, unwound product for several flavors, using defaults

```
% upd addproduct foo v1_0 -f IRIX:SunOS:OSF1
```

This example is similar to the first, but shows declaring the product on *fnkits* for three different flavors at the `-1` level. `upd addproduct` gets run three times, once for each flavor.

Add undeclared, unwound product

```
% upd addproduct foo v1_0 -P -2 -m foo.table -M ups \  
-r /path/to/foo/prodrootdir -U /path/to/foo/prodrootdir/ups
```

This time the product has not been declared to a local database. Therefore **UPS/UPD** cannot determine where to find the product root directory (**-r**), the table file (**-m** and **-M**) or the `ups` directory (**-U**) on the local node. Again, the flavor is the **-2** flavor level of the local machine. We include **-P** to ensure that no instance declared in the database(s) can be selected in place of the one specified.

Add locally declared tar file

```
% upd addproduct foo v1_0 -2 -T /tmp/foo_v1_0_SunOS+5.tar
```

For this example, we assume the product instance was declared to a local **UPS** database before the tar file was created. The tar file includes the entire structure under the product root directory. **UPD** picks up the pre-made tar file from the local machine in `/tmp/foo_v1_0_SunOS+5.tar` (specified using `-T`), adds it to *fnkits* (no `-h`), and declares it with the `-2` flavor level of the local machine, no chain, and no qualifiers.

Add undeclared product with external ups directory but no table file

```
% upd addproduct footwo v1_0 -P0 -h dist_node.fnal.gov \
-T /tmp/footwo_v1_0_NULL.tar -U /local/path/to/ups/dir
```

UPD relies solely on information supplied on the command line to execute this command (`-P`). It picks up the tar file (path given via `-T`) of product **footwo** v1_0, flavor NULL (`-0`). No table file is specified (no `-m` or `-M`), therefore **UPD** doesn't look for one. This product has a `ups` directory external to the product root directory (given via `-U`). **UPD** adds the product to the (fictional) node *dist_node.fnal.gov* (given via `-h`).

Add undeclared product consisting only of a table file

```
% upd addproduct foothree v1_0 -Pf IRIX -m foothree.table \
-M /local/path/to/table/file
```

UPD relies solely on information supplied on the command line to execute this command (`-P`). It picks up the product, **foothree** v1_0 of flavor IRIX, which consists only of a table file (it may be a bundled product). The `-m` and `-M` options are included to specify the table file name and location, there is no product root directory, and thus no `-r`. **UPD** adds it to *fnkits* and declares it with the flavor IRIX, no chain and no qualifiers.

The system returns a notice message saying there is no product root directory. This is correct behavior, and is expected.

24.2 upd cloneproduct

The **upd cloneproduct** command creates a new product instance (the target instance) on a distribution node by copying one that is already there (the source instance) and changing one or more of its identifying elements.

24.2.1 Command Syntax

```
% upd cloneproduct <flavor_option> [<source_options>] <product>
\ [<version>] -G "<target_options>"
```

24.2.2 All Valid Options

- ? ("-?" for **cs**) Prints command description and option usage information to screen
- f <flavorList> Described below under “The flavor options”.
- G "<options>" Specifies options to be passed to the **ups declare** command on the distribution node for the target instance; see below
- h <host> Specifies product distribution host to which you are adding the product; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
 - a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the **ups.cgi** script (e.g., *http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi*)
- i Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one, operation continues to the next.
- q <qualifierList> Finds source product instance on distribution node with the specified qualifiers (required and/or optional)
- v(vvv) Prints out extra debugging information

The flavor options

Flavor may be specified using **-f**, or using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with **-f** or with each other.

- f <flavorList> Specifies flavor of source product on the distribution node. If multiple values specified, **upd cloneproduct** is run once for each flavor.
- 0 Specifies flavor of source product on the distribution node as NULL; equivalent to **-f NULL**

- 1 Specifies flavor of source product on the distribution node as the OS value of local machine up to the generic OS (e.g., equivalent to **-f SunOS**)
- 2 Specifies flavor of source product on the distribution node as OS value of local machine up to the version (e.g., equivalent to **-f SunOS+5**)
- 3 Specifies flavor of source product on the distribution node upto the release of the version (e.g., equivalent to **-f SunOS+5.6**)
- 4 Specifies flavor of source product on the distribution node upto the patch number of the release (e.g., equivalent to **-f SunOS+5.6.2**)

24.2.3 Options Valid with -G

In order to distinguish the target product instance from the source, the declarations for the two instances must differ by at least one instance-identifying element. The **-G** option provides the means to specify the target instance identifiers; it takes a list of **ups declare** command line elements as an argument. Any identifier not specified via **-G** retains the value of the source instance. The elements valid for use with **-G** include **<product>**, **<version>** and the following subset of the **ups declare** options:

-A <nodeList>, **-c**, **-d**, **-D <origin>**, **-f <flavor>**, **-g <chainName>**, **-n**, **-o**, **-O "<flagList>"**, **-p "<description>"**, **-q <qualifierList>**, **-t**, **-z <databaseList>**, **-0**, **-1**, **-2**, **-3**, **-4**

See section 23.5 *ups declare* for details on each option. If the argument to **-G** includes the product version, the product name must be included ahead of the version; the first unflagged element is always interpreted as the product name and the second as the version.

24.2.4 upd cloneproduct Example

```
% upd cloneproduct -f NULL jfc v1_0 -G "-f CYGWIN32_NT"
```

In this example, **UPD** finds the (OS-independent) product **jfc** version **v1_0** of flavor **NULL**, and clones a new instance with all identifiers the same except for the flavor. The new instance has the flavor **CYGWIN32_NT**, for NT users. Going to **CYGWIN32_NT** presents a problem, so we provide some further explanation:

This product contains **java** classes. For **java** to dynamically load class libraries, it looks in an environment variable called CLASSPATH which contains the directories with the **java** classes you want to use. On UNIX, the directories in CLASSPATH need to be colon (:) separated, but in CYGWIN, they need to be semi-colon (;) separated.

We've handled this by setting a delimiter variable in the product's table file. One instance in the table file is defined for NULL (using colon delimiters), and one is for CYGWIN32_NT (using semi-colons). For example, in the table file under the SETUP action for Flavor=NULL we have:

```
envPrepend (CLASSPATH, ${UPS_PROD_DIR}/swingall.jar, ":")
```

and under Flavor=CYGWIN32_NT, it is changed to:

```
envPrepend (CLASSPATH, $jfc_cpath, ";")
```

Everything in the two product instances is exactly the same, except the delimiters.

24.3 upd delproduct

The **upd delproduct** command deletes a product declaration from a distribution database. It also removes any associated tar file, table file and/or ups directory. The product subdirectory itself does not get deleted.

24.3.1 Command Syntax

```
% upd delproduct -f <flavor_option> [<other_options>] <product>  
  \ <version>
```

24.3.2 Commonly Used Options

See section 24.3.3 *All Valid Options* for descriptions of each option.

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (together with one of **-0**, **-1**, **-2**, **-3**, **-4**)

-h <host>

-q <qualifierList>

24.3.3 All Valid Options

-? ("-" for csh) Prints command description and option usage information to screen

- f <flavorList>** Described below under “The flavor options”.
- h <host>** Specifies product distribution host from which you are deleting the product; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
 - a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi`)
- H <flavorList>** Described below under “The flavor options”.
- i** Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one, operation continues to the next.
- O "<flags>"** Sets the value of `$UPS_OPTIONS` to **<flags>**. This value would get passed to the host's `updconfig` file. This is not used on *fnkits* (see section 24.1.5 *Adding Products to fnkits.fnal.gov*).
- q <qualifierList>** Finds product instance on distribution node with the specified qualifiers (required and/or optional)
- s** Lists what command would do; but does not execute the command
- v(vvv)** Prints out extra debugging information.

The flavor options

Flavor may be specified using **-f**, using **-H** in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).

- f <flavorList>** Finds product instance of specified flavor on distribution node. If multiple values specified, **upd delproduct** is run once for each flavor.
- H <flavorList>** Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families.
 Must be used with one of **-0**, **-1**, **-2**, **-3**, **-4**, to specify flavor absolutely. Used this way, it finds product instance of specified flavor on distribution node; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd delproduct** is run once for each flavor, according to the accompanying number option.

- 0 Finds product instance of NULL flavor on distribution node; equivalent to **-f NULL**
- 1 Finds product instance (on distribution node) of local OS flavor, specified up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.
- 2 Finds product instance (on distribution node) of local OS flavor, specified up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.
- 3 Finds product instance (on distribution node) of local OS flavor, specified up to the release of the version; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4 Finds product instance (on distribution node) of local OS flavor, specified up to the patch of the release; e.g., equivalent to **-f SunOS+5.6.2**; if given together with **-H IRIX+6.2.1**, flavor is then specified as IRIX+6.2.1.

24.3.4 upd delproduct Example

```
% upd delproduct foo v1_0 -cf SunOS+5
```

This command deletes the product **foo** v1_0 declared as “current” for the flavor SunOS+5. (Since the version is specified, the **-c** is unnecessary, but harmless.) The command syntax is the same whether the product is in archived format, is unwound or consists of just a table file.

24.4 upd depend

The **upd depend** command executes **ups depend** on a product distribution database. The **ups depend** command lists product dependencies of the specified product instance(s) as declared locally. See section 23.6 *ups depend* for more information and examples.

upd install runs this command internally to determine what dependencies to install with the requested product. Product installers can use it to see what products **upd install** would distribute.

24.4.1 Command Syntax

```
% upd depend [-h <host>] [<ups_depend_options>] <product> \
    [<version>]
```

24.4.2 Options

The **upd depend** command uses all the same options as **ups depend**, plus **-h** (described below). See section 23.6 *ups depend* for the list of options and their descriptions. The difference in the options' functionality is that they apply to the distribution node rather than to the local node.

-h <host> Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:

- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
- a host and Webserver port number (e.g., *fred.sub.domain:8080*)
- a full URL to the *ups.cgi* cgi script (e.g., *http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi*)

24.4.3 upd depend Examples

```
% upd depend exmh
```

This example lists all the dependencies declared for the default instance of **exmh** on the default host *fnkits*:

```
exmh v2_0_2 -f NULL -z /ftp/upsdb -g current
|__expect v5_25 -f SunOS+5 -z /ftp/upsdb -g current
|  |__tk v8_0_2 -f SunOS+5 -z /ftp/upsdb
|    |__tcl v8_0_2 -f SunOS+5 -z /ftp/upsdb
|__mh v6_8_3c -f SunOS+5 -z /ftp/upsdb -g current
|  |__mailtools v2_3 -f NULL -z /ftp/upsdb -g current
|__mimetools v2_7a -f SunOS+5 -z /ftp/upsdb -g current
|__glimpse v3_0a -f SunOS+5 -z /ftp/upsdb -g current
|__www v3_0 -f NULL -z /ftp/upsdb -g current
|  |__lynx v2_8_1 -f SunOS+5 -z /ftp/upsdb -g current
|__ispell v3_1b -f SunOS+5 -z /ftp/upsdb -g current
```

To specify a different host, use the **-h** option, e.g.,

```
% upd depend -h dist_node exmh
```

If a chain rather than a version number is used to specify the instance (as is the case for the default “current” instance), then the chain appears in the output line for the product (notice the `-g current` in the first line); otherwise the chain is not listed.

We refer you to section 23.6 *ups depend* for more examples, as the two commands use the same syntax and options (except for `-h`) and produce similar output.

24.5 upd exist

The **upd exist** command runs **ups exist** on a product distribution node. The **ups exist** command is used to test whether a **setup** command issued on the local machine with the same command line elements is likely to succeed. See section 23.7 *ups exist* for more information and examples.

upd exist can be used to test for the existence of a particular product instance on a distribution node, prior to installing it. It can be used whether the distribution database is “live” or in archive format.

24.5.1 Command Syntax

```
% upd exist [-h <host>] [<ups_exist_options>] <product> \
  [<version>]
```

24.5.2 Options

The **upd exist** command uses all the same options as **ups exist**, plus `-h` (described below). See section 23.7 *ups exist* for the list of options and their descriptions. The difference in the options’ functionality is that they apply to the distribution node rather than to the local node.

-h <host>	Specifies product distribution host; the default is <i>fnkits.fnal.gov</i> . Usually just the plain host name is required, however the <code>-h</code> option can always specify any of the following: <ul style="list-style-type: none">- a plain host name; (e.g., <i>fred.sub.domain</i>, or just <i>fred</i> if <i>sub.domain</i> is <i>fnal.gov</i>)- a host and Webserver port number (e.g., <i>fred.sub.domain:8080</i>)- a full URL to the <code>ups.cgi</code> cgi script (e.g., <code>http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi</code>)
------------------------	--

24.5.3 upd exist Examples

This command is rarely used from the command line, and is more useful in scripts where a failed setup could cause the script to abort. When issued from the command line, it returns no output if the command succeeds.

In the C shell family **upd exist** sets the `$status` variable to 0 if it was able to create the temporary file, or to 1 for error. In the Bourne shell family, it sets the `$?` variable similarly. As an example, we can run **upd list** and find that there is a current instance of the product **tex** for the flavor IRIX+6 but not for IRIX+6.2. Running **upd exist** for each flavor, we see that the variables get set accordingly:

For the C shell family:

```
% upd exist tex -f IRIX+6; echo $status
```

0

```
% upd exist tex -f IRIX+6.2; echo $status
```

1

For the Bourne shell family:

```
$ upd exist tex -f IRIX+6; echo $?
```

0

```
$ upd exist tex -f IRIX+6.2; echo $?
```

1

24.6 upd fetch

The **upd fetch** command performs either of the following functions:

- If **-J** is used, it retrieves a single file or directory from a product distribution database, and downloads it to the user node, placing it relative to the current working directory.
- If **-J** is not used, returns a recursive list of directories and files that are available for retrieval from the product distribution node. Warning: When using a live distribution database, this list may be very long.

The **upd fetch** command cannot retrieve files from within a tar file. In an archive distribution database, typically the only files provided in the appropriate format are the **README** and **INSTALL_NOTE** files, the **ups** subdirectory files, and the table and version files.

This command is used internally by **UPD**, and only rarely run manually.

24.6.1 Command Syntax

```
% upd fetch [<options>] [-J fileName] <product> [<version>]
```

24.6.2 Commonly Used Options

See section 24.6.3 *All Valid Options* for descriptions of each option.

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (alone or together with one of **-0**, **-1**, **-2**, **-3**, **-4**)

-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**

-h <host>

-J <fileName>
-q <qualifierList>

24.6.3 All Valid Options

-? ("-?" for **csb**) Prints command description and option usage information to screen

-c Finds product instance chained to "current" on the distribution node

-d Finds product instance chained to "development" on the distribution node

-f <flavorList> Described below under "The flavor options".

-g <chainName> Finds product instance chained to <chainName> on the distribution node; can be a list of chain names

- h <host>** Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/ome/dir/ups.cgi`)
- H <flavorList>** Described below under “The flavor options”.
- i** Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one, operation continues to the next.
- J <fileName>** Specifies an individual file to fetch from the distribution node (it does not accept a list of files, however you can retrieve a file with a colon in its name). The argument **@table_file** is valid for specifying the table file.
- n** Finds product instance chained to “new” on the distribution node
- o** Finds product instance chained to “old” on the distribution node
- q <qualifierList>** Finds product instance on distribution node with the specified qualifiers (required and/or optional)
- s** Lists what command would do; but does not execute the command
- t** Finds product instance chained to “test” on the distribution node
- v(vvv)** Prints out extra debugging information.

The flavor options

Flavor may be specified using **-f**, using **-H** by itself or in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).

- f <flavorList>** Finds product instance of specified flavor on distribution node. If multiple values specified, **update** is run once for each flavor.

- H <flavorList>** Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families.
- Can be used alone (without an accompanying number option). In this case, the best match instance of the flavor family on the distribution node is picked.
- Can also be used with one of **-0**, **-1**, **-2**, **-3**, **-4**, to specify flavor absolutely. Used this way, it finds product instance of specified flavor on distribution node; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd fetch** is run once for each flavor, according to the accompanying number option.
- 0** Finds product instance of NULL flavor on distribution node; equivalent to **-f NULL**
- 1** Finds product instance (on distribution node) of local OS flavor, specified up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.
- 2** Finds product instance (on distribution node) of local OS flavor, specified up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.
- 3** Finds product instance (on distribution node) of local OS flavor, specified up to the release of the version; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4** Finds product instance (on distribution node) of local OS flavor, specified up to the patch number of the release; e.g., equivalent to **-f SunOS+5.6.2**; if given together with **-H IRIX+6.2.1**, flavor is then specified as IRIX+6.2.1.

24.6.4 upd fetch Examples

Output a list of files and directories

```
% upd fetch teledata
```

This first example illustrates the command output when the **-J** option is omitted on a request to a live distribution database. Nothing actually is retrieved when **-J** is absent. The output is a recursive list of directories and files that are available for individual retrieval (list edited for brevity):

```
Listing of table_dir [/ftp/products/teledata/v1_0/NULL]:
```



```

total 26442
drwxrwx---    3 updadmin upd          512 May 12 1999
teledata_v1_0_NULL
-rw-rw-r--    1 updadmin upd          712 May 12 1999
teledata_v1_0_NULL.table
-rw-rw----    1 updadmin upd          712 Jan 28 1999
teledata_v1_0_NULL.table.old
-rw-rw----    1 updadmin upd        13516800 May 12 1999
teledata_v1_0_NULL.tar
-rw-rw----    1 updadmin upd          9728 May 12 1999
teledata_v1_0_NULL.ups.tar

teledata_v1_0_NULL:
total 12
-rw-r--r--    1 updadmin upd        4513 Feb  9 1999 README
drwxr-xr-x    2 updadmin upd          512 Jan 28 1999 ups

teledata_v1_0_NULL/ups:
total 12
lrwxrwxrwx    1 updadmin upd           9 May 12 1999
INSTALL_NOTE -> ../README
-rwxr-xr-x    1 updadmin upd          541 May 12 1999
configure
-rwxr-xr-x    1 updadmin upd          568 May 12 1999 current
-rw-r--r--    1 updadmin upd          784 Jan  5 1999
teledata.table
-rwxr-xr-x    1 updadmin upd          209 May 12 1999
unconfigure
-rwxr-xr-x    1 updadmin upd          212 May 12 1999
uncurrent

Listing                               of                               @ups_dir
[/ftp/products/teledata/v1_0/NULL/teledata_v1_0_NULL/ups]:
total 12
...
```

```

Listing                                of                                @prod_dir
[/ftp/products/teledata/v1_0/NULL/teledata_v1_0_NULL]:
total 12
-rw-r--r--    1 updadmin upd          4513 Feb  9  1999 README
drwxr-xr-x    2 updadmin upd          512 Jan 28  1999 ups

ups:
total 12
...

```

Retrieve a file

```
% upd fetch -f IRIX+6 -J README tex v3_14159
```

This example shows the retrieval of a `README` file (`-J README`) for the product `tex`. We specifically request the `README` pertaining to the flavor `IRIX+6` for the product version `v3_14159`. The file will be copied to the current working directory. The system provides some informational output:

```

informational: transferred README
from
fnkits.fnal.gov:/ftp/products/tex/v3_14159/IRIX+6/tex_v3_141
59_IRIX+6
to ./README

```

Retrieve the table files for several flavors of a product

```
% upd fetch -H SunOS+5:IRIX+6:Linux+2:OSF1+V4 -J @table_file \
tex v3_14159
```

This example retrieves the table file(s) for the best match product instances of `tex v3_14159` for the listed flavor families. Depending on how the product was configured, the same table file may be used for all, or they may be separate files. The file(s) will be copied to the current working directory.

24.7 upd get

The `upd get` command runs `ups get` on a product distribution node. Currently they can only be used with the `-F` option. `upd get -F` lists any files on the distribution node which are to be distributed with the specified product instance(s) and which are maintained outside of the product root directory. The list does not include table files, for which the location is maintained in the version file. See section 23.9 *ups get* for more information and examples.

The **ups get** and **upd get** commands were designed primarily for use by **UPD**, which calls it internally. As such they are rarely used outside of that context. In a future release, **ups get** may acquire additional functions.

24.7.1 Command Syntax

```
% upd get -F [-h <host>] [<ups_get_options>] <product>
  [<version>]
```

24.7.2 Options

The **upd get** command uses all the same options as **ups get**, plus **-h** (described below). See section 23.9 *ups get* for the list of options and their descriptions. The difference in the options' functionality is that they apply to the distribution node rather than to the local node.

-h <host> Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:

- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
- a host and Webserver port number (e.g., *fred.sub.domain:8080*)
- a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)

24.8 upd install

The **upd install** command retrieves a product instance and by default its dependencies, as needed, from a product distribution node. It installs the retrieved instances on the local node, declares them to a local **UPS** database, and optionally runs commands needed to resolve dependencies.

24.8.1 Command Syntax

```
% upd install [<options>] <product> [<version>]
```

24.8.2 Commonly Used Options

See section 24.8.3 *All Valid Options* for descriptions of each option.

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (alone or together with one of **-0**, **-1**, **-2**, **-3**, **-4**)

-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**

-G "<options>"

-h <host>

-i

-I

-j

-q <qualifierList>

-R

-X

-z <databaseList>

24.8.3 All Valid Options

-? ("-" for *cs*h) Prints command description and option usage information to screen

-c Finds product instance chained to “current” on the distribution node

-C Prevents execution of the CONFIGURE action during the initial local product declaration run by **upd install**. It also prevents execution of any action that would otherwise be called by options listed in the **-G** argument (e.g., **-G "-c"** normally would call CURRENT, but in this case acts like **-G "-c -C"**, which would not call CURRENT).

-d Finds product instance chained to “development” on the distribution node

-f <flavorList> Described below under “The flavor options”.

-g <chainName> Finds product instance chained to **<chainName>** on the distribution node; can be a list of chain names

-G "<options>" Specifies options to be passed to the local **ups declare** command; see below

-h <host> Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:

- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
- a host and Webserver port number (e.g., *fred.sub.domain:8080*)
- a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)

- H <flavorList>** Described below under “The flavor options”.
- i** Ignores errors; if a dependency can't be downloaded or installed locally, operation continues.
- I** Unwinds product tar file on the local node after transmission (default is to unwind during transmission). \$TMPDIR is used as the tar file destination. Use this if your network is unreliable. Note that you need twice the disk space for installs.
- j** Ignores dependencies, installs specified product only
- m <tableFileName>** Specifies table file name for the product instance on the distribution node
- M <tableFileDir>** Specifies table file directory for the product instance on the distribution node
- n** Finds product instance chained to “new” on the distribution node
- o** Finds product instance chained to “old” on the distribution node
- O "<flags>"** Sets the value of \$UPS_OPTIONS to **<flags>**. You'd use this if your `updconfig` file had corresponding stanzas.
- q <qualifierList>** Finds product instance on distribution node with the specified qualifiers (required and/or optional)
- r <prodRootDir>** Specifies the product root directory for the product instance on the distribution node
- R** Downloads and installs only the required dependencies of the specified product.
- s** Lists what command would do; but does not execute the command
- t** Finds product instance chained to “test” on the distribution node
- U <upsDir>** Specifies location of `ups` directory; default value is **ups**
- v(vvv)** Prints out extra debugging information.

- X** Executes the generated **ups declare** commands needed to resolve dependencies. If **-X** is not included, **ups install** echoes the commands to the screen.
- z <databaseList>** Specifies the local database(s) in which product and its dependencies can be installed. See section 27.1 *Database Selection Algorithm* for info on how the database is chosen.

If **-z** is specified as a normal option to **upd install**, then it determines the database in which to look for the **UPD** configuration; if not, then **\$PRODUCTS** is used for this purpose. If specified within **-G** construction, then it gets handed to **ups declare** and is used for the second declaration. It may be specified in both places, but if not done with appropriate forethought, this can lead to errors.

The flavor options

Flavor may be specified using **-f**, using **-H** by itself or in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).

- f <flavorList>** Finds product instance of specified flavor on distribution node, and sets flavor by default on local node. If multiple values specified, **upd install** is run once for each flavor. Avoid using **-f** except in a **-G** argument string, because it can cause flavor mismatches between main product and dependencies.
- H <flavorList>** Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families.

Can be used alone (without an accompanying number option). In this case, the best match instance of the flavor family on the distribution node is picked.

Can also be used with one of **-0**, **-1**, **-2**, **-3**, **-4**, to specify flavor absolutely. Used this way, it finds product instance of specified flavor on distribution node; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd install** is run once for each flavor, according to the accompanying number option.
- 0** Finds product instance of NULL flavor on distribution node; equivalent to **-f NULL**

- 1 Finds product instance (on distribution node) of local OS flavor, specified up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.
- 2 Finds product instance (on distribution node) of local OS flavor, specified up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.
- 3 Finds product instance (on distribution node) of local OS flavor, specified up to the release of the version; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4 Finds product instance (on distribution node) of local OS flavor, specified up to the patch number of the release; e.g., equivalent to **-f SunOS+5.6.2**; if given together with **-H IRIX+6.2.1**, flavor is then specified as IRIX+6.2.1.

24.8.4 Options Valid with -G

The **-G** option is provided to allow you to pass **UPS** options to the **ups declare** command. The **upd install** command first declares the product without referencing **-G**, then does a second declaration with this information. Any identifier not specified via **-G** retains its previous value. The elements valid for use with **-G** include **<product>**, **<version>** and the following subset of the **ups declare** options:

-A <nodeList>, **-c**, **-d**, **-D <origin>**, **-f <flavor>**, **-g <chainName>**, **-n**, **-o**, **-O "<flagList>"**, **-p "<description>"**, **-q <qualifierList>**, **-t**, **-z <databaseList>**, **-0**, **-1**, **-2**, **-3**, **-4**

If you include no instance identifiers within **-G** (i.e., exclude flavor, **-q**, and **-z**) so that the second declaration is just modifying the previously declared instance (e.g., setting a chain), then the options **-A**, **-D** and **-p** are ignored.

See section 23.5 *ups declare* for details on each option. If the argument to the **-G** option includes the product version, it must also include the product name ahead of the version; the first unflagged element is always interpreted as the product name and the second as the version.



Note that options included here apply to the product's dependencies as well as to the main product. Make sure that you don't assign chains or flavors to dependencies required by other products in such a way as to make those products break.

24.8.5 More Detailed Description

The **upd install** command performs the following series of functions:

- retrieves the specified product instance, and by default its dependencies, from a distribution node
- installs the product on the user node according to the node's **UPD** configuration
- unwinds the product if transferred in tar format
- declares the product to a local database (by calling **ups declare**); the database specified in the node's **UPD** configuration is the default, but it can be overridden on the command line
- prints to screen the commands you will need to issue in order to resolve dependencies (Usually these are chain declarations. Most of the time dependencies are based on chains not versions.)

Internal Processes

The **upd install** command operates by making a series of network connections to the server. All calls are made from the client system to the distribution server, who reports back results on the same data channel:

- The Web server on the distribution node is called, and `ups.cgi` is used to determine if the product instance in question exists on the server, what its dependencies are. A call to the local **UPS** determines whether the product and its dependencies exist on the user node. For the product itself and for each dependency not found on the user node, the remaining steps are taken:
- The Web server is called, and `ups.cgi` is used to determine particular details of the product on the distribution node (e.g., archive file location, product root directory, `ups` directory, and so on). If no archive file location is given, **UPD** manufactures a tar file that should work, assuming the **FTP** server can make a tar file of directories on the fly.¹ The tar file gets named according to the convention:
`ftp://host/$UPS_PROD_DIR/.tar` (a “.” for the path, followed by “.tar”).
- The **FTP** server is used to transfer and unwind the archive file, an archive of the `ups` directory, and the table file for the product.
- **UPD** declares the product on the local system.

24.8.6 upd install Examples

Default installation

```
% upd install tex
```

In this example, **UPD** downloads the current instance of the product **tex** (for the flavor of the local machine) from the default distribution node *fnkits*, and installs and declares it on the local machine using the defaults in place. Dependencies, as needed, also get installed.

Install only default instance into specified database and make “current”

```
% upd install -z $MYDB -G "-c" -j tex
```

```
% upd install -z /path/to/my/database -G "-g current" -j tex
```

1. A **WU-FTP** compatible **FTP** server is used to make tar files “on the fly”.

These two examples are equivalent assuming \$MYDB is set to the path shown. They install only the product **tex** (**-j** specifies no dependencies) and include the **-z** option to specify the target database. The product **tex** is constrained to exist in the specified database, and it gets chained to “current”.

Install multiple flavors of a product

```
% upd install perl v5_005 -H CYGWIN32_NT:Linux+2:SunOS+5:IRIX+6 -C
```

UPD retrieves the instances of **perl** version v5_005 for the listed flavors, and all dependencies (correctly matched for flavor), as needed. If **-f** were used in place of **-H** here, all the dependencies installed would be of the best match flavor to the local machine, and most or all product instances wouldn’t work right.

The **-C** option prevents execution of the **CONFIGURE** action, which may be OS-specific. In this case, the installer needs to login to one machine of each flavor in the cluster and manually run **ups configure** for **perl**. **ups configure** is described in section 23.3 *ups configure*.

Install product plus required dependencies

```
% upd install exmh v2_0_2 -h dist_node -R
```

This command installs the product **exmh** version v2_0_2 for the best match flavor of the local machine. It uses the distribution node *dist_node*. The option **-R** causes only the product and its required dependencies (as needed) to get installed.

Install a product from a CD-ROM

A distribution database can be on CD-ROM. See <http://www.fnal.gov/docs/products/ups/ReferenceManual/misc/cdrom.html> for more information. To install a product from CD-ROM, first insert the CD-ROM and run the **mount** command:

```
% mount -t iso9660 /dev/cdrom /path/to/db/on/cdrom
```

Then to install a product from the CD-ROM, run the **upd install** command normally, but include the **-h** option pointing to the database on the CD-ROM as shown:

```
% upd install -h file://localhost/path/to/db/on/cdrom ...
```

The Linux CD-ROM images created by the Computing Division mount the distribution database on the CD-ROM to `/tmp/rhimage/products/db`. So, for example, the commands would look like:

```
% mount -t iso9660 /dev/cdrom /tmp/rhimage
```

```
% upd install -h file://localhost/tmp/rhimage/products/db
<product> <version> [-G <options>] ...
```

24.9 upd list

The **upd list** command performs **ups list** on a distribution database.

The **ups list** command returns information about the declared product instances in a local **UPS** database. Two output styles are provided: a formatted one that is easy for users to read, and a condensed one for parsing by a subsequent command or a script.

See 23.11 *ups list* for more information and examples.

24.9.1 Command Syntax

```
% upd list [-h <host>] [<ups_list_options>] [<product>] \
[<version>]
```

24.9.2 Options

The **upd list** command uses all the same options as **ups list**, plus **-h** (described below). See section 23.11 *ups list* for the list of options and their descriptions. The difference in the options' functionality is that they apply to the distribution node rather than to the local node.

-h <host>	Specifies product distribution host; the default is <i>fnkits.fnal.gov</i> . Usually just the plain host name is required, however the -h option can always specify any of the following: <ul style="list-style-type: none">- a plain host name; (e.g., <i>fred.sub.domain</i>, or just <i>fred</i> if <i>sub.domain</i> is <i>fnal.gov</i>)- a host and Webserver port number (e.g., <i>fred.sub.domain:8080</i>)- a full URL to the <code>ups.cgi</code> cgi script (e.g., <code>http://fred.sub.domain:8080/cgi-bin/come/dir/ups.cgi</code>)
------------------------	--

24.9.3 upd list Examples

We refer you to section 23.11 *ups list* for many examples. Simply substitute **upd list** for **ups list**, and the commands perform the same function, but on a distribution node. Use the **-h <host>** option to specify a distribution node different from the default *fnkits.fnal.gov*, e.g., the command:

```
% upd list -aK+ -h dist_node emacs
```

This requests the standard output fields for all instances (**-a** for all) of **emacs**, from the distribution node *dist_node*, using the condensed output format (**-K+**):

```
"emacs" "v19_30a" "AIX+3" "" ""
"emacs" "v19_30a" "IRIX+5" "" ""
"emacs" "v19_30a" "SunOS+5" "" ""
...
"emacs" "v19_34b" "SunOS+5" "" "current"
```

24.10 upd modproduct

The **upd modproduct** command is used to modify a product instance that already exists in a distribution database. It allows you to replace a table file or **ups** directory, or to add or change chain information for the product. To modify any other data, you must delete the product or tar file from the distribution database and add it again with the new values (see sections 24.3 *upd delproduct* and 24.1 *upd addproduct*). **upd modproduct** does not modify product tar files.

24.10.1 Command Syntax

For replacing a table file

```
% upd modproduct <flavor_option> -m <tableFileName> \
  [-M <tableFileDir>] [<other_options>] <product> <version>
```

Note: You must include the **-m** option specifying the table file name, as there is no default. You must also include **-M** if the table file is not in the current directory.

For replacing a ups directory

```
% upd modproduct <flavor_option> -U <upsDir> [<other_options>]\
  <product> <version>
```

For adding or changing a chain

```
% upd modproduct <flavor_option> <chain_option>
  [<other_options>]\ <product> <version>
```

24.10.2 Commonly Used Options

See section 24.10.3 *All Valid Options* for descriptions of each option.

For replacing a table file or ups directory

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (together with one of **-0**, **-1**, **-2**, **-3**, **-4**)
-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**
-h <host>
-m <tableFileName>
-M <tableFileDir>
-q <qualifierList>
-U <upsDir>

For adding or changing a chain

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (together with one of **-0**, **-1**, **-2**, **-3**, **-4**)
-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**
-h <host>
-q <qualifierList>

24.10.3 All Valid Options

-? (" -?" for csh) Prints command description and option usage information to screen

-c When adding a chain, assigns the “current” chain to the product instance on the distribution node; when replacing a component, ignored.

-d When adding a chain, assigns the “development” chain to the product instance on the distribution node; when replacing a component, ignored.

-f <flavorList> Described below under “The flavor options”.

-g <chainName> When adding a chain, assigns the **<chainName>** chain to the product instance on the distribution node; when replacing a component, ignored. Can be a list of chain names.

- h <host>** Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)
- H <flavorList>** Described below under “The flavor options”.
- i** Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one, operation continues to the next.
- m <tableFileName>** When uploading a table file and/or `ups` directory containing a new table file, specifies local table file name
- M <tableFileDir>** When uploading a table file and/or `ups` directory containing a new table file, specifies local table file directory
- n** When adding a chain, assigns the “new” chain to the product instance on the distribution node; when replacing a component, ignored.
- o** When adding a chain, assigns the “old” chain to the product instance on the distribution node; when replacing a component, ignored.
- q <qualifierList>** Finds product instance on distribution node with the specified qualifiers (required and/or optional)
- s** Lists what command would do; but does not execute the command
- t** When adding a chain, assigns the “test” chain to the product instance on the distribution node; when replacing a component, ignored.
- U <upsDir>** When uploading a `ups` directory, specifies its location on local node; default value is **ups** (relative to the product root directory)
- v(vvv)** Prints out extra debugging information.

The flavor options

Flavor may be specified using **-f**, using **-H** by itself or in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).

- f <flavorList>** Finds product instance of specified flavor on distribution node. If multiple values specified, **upd modproduct** is run once for each flavor.
- H <flavorList>** Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families. If multiple values specified, **upd modproduct** is run once for each value.

Must be used with any of **-0**, **-1**, **-2**, **-3**, **-4**; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd modproduct** is run once for each flavor, according to the accompanying number option.
- 0** Specifies flavor as NULL; equivalent to **-f NULL**
- 1** Specifies flavor as OS value up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.
- 2** Specifies flavor for product instance on local and distribution nodes as OS value up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.
- 3** Specifies flavor for product instance on local and distribution nodes upto the release; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4** Specifies flavor for product instance on local and distribution nodes upto the patch number of the release; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.

24.10.4 More Detailed Description

Note that **upd modproduct** cannot query the local **UPS** database to find information the way **upd addproduct** can; all necessary information must be specified on the command line.

A product instance on a distribution node generally has at most one chain associated with it at any time.¹ Whenever you change a chain with **upd modproduct**, you automatically delete any and all previously assigned chain or chains.

1. A product instance can have multiple chains if they are declared together in the same command (e.g., **upd modproduct -g test:current ...**).

When you use **upd modproduct** to replace a **ups** directory:

- If the **ups** directory contains a newer table file that should replace the old one, include the **-m** and **-M** options in the command.
- Do not make a tar file of the **ups** directory on your local machine.

24.10.5 upd modproduct Examples

Replace a table file

```
% upd modproduct foo v1_0 -2 -m v1_0.table -M \
/local/path/to/foo/ups/dir
```

For this example, we assume that a new table file has replaced the old one in the product instance's local **ups** directory, and it must now be added to the distribution node. In this **upd modproduct** command:

- we identify the product instance associated with the table file: **foo v1_0**, of the level **-2** flavor of the local machine, in **\$PRODUCTS**
- **-m v1_0.table** gives the name of the new table file
- **-M /local/path/to/foo/ups/dir** gives the table file directory. Even though the table file is in one of the default locations, the location must be specified unless you issue the command from the directory specified by **-M**. The **upd modproduct** command doesn't query **UPS** for the information.

Replace a ups directory

```
% upd modproduct foofour v1_0 -f SunOS -h dist_node -U \
/local/path/to/ups/dir
```

In this example, the **upd modproduct** command is used with the **-U** option to replace a product instance's **ups** directory. We illustrate with a product called **foofour v1_0**, flavor **SunOS**, and no qualifiers. We distribute it to *dist_node* using **-h dist_node** (the distribution database is determined by the **UPD** configuration on *dist_node*). It doesn't matter whether the product instance is declared to a **UPS** database listed in **\$PRODUCTS**, since **upd modproduct** won't query the database anyway. Regardless of its location, the **ups** directory location must be fully specified.

Declare a chain

```
% upd modproduct foo v1_0 -t -i -f
SunOS+5:IRIX+6:Linux+2:OSF1+V4
```

For this example, we assume that the product **foo** v1_0 has no chain in its **KITS** declarations (default distribution host used here). In this command we declare a “test” chain for several flavors using the **-t** option (**-g test** would work too). We include **-i** in case any of the flavors doesn’t exist; the command will proceed to the next flavor.

Change a declared chain

```
% upd modproduct foo v1_0 -c -f SunOS+5:IRIX+6:Linux+2:OSF1+V4
```

This command changes the “test” chains for **foo** in the above example to “current”. This removes the test chains. Again, **-i** allows the command to proceed in case of errors.

Remove a chain

```
% upd modproduct foo v1_0 -f SunOS+5 -g :
```

This command removes a chain on the specified instance. It doesn’t assign a new one, nor does it assign the existing chain to a different instance. This often generates warnings, but it works and causes no database problems.

24.11 upd parent

The **upd parent** command executes **ups parent** on a product distribution database. The **ups parent** command can be used to determine which products depend on the specified product instance(s) as declared in the (local) database. See section 23.13 *ups parent* for more information and examples.

24.11.1 Command Syntax

```
% upd parent [-h <host>] [<ups_parent_options>] <product> \
    [<version>]
```

24.11.2 Options

The **upd parent** command uses all the same options as **ups depend**, plus **-h** (described below). See section 23.13 *ups parent* for the list of options and their descriptions. The difference in the options’ functionality is that they apply to the distribution node rather than to the local node.

-h <host> Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:

- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
- a host and Webserver port number (e.g., *fred.sub.domain:8080*)
- a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)

24.11.3 upd parent Examples

We refer you to section 23.13 *ups parent* for examples, as the two commands use the same syntax and options (except for **-h**) and produce similar output.

24.12 upd reproduct

The **upd reproduct** command is equivalent to a **upd delproduct** followed by a **upd addproduct**. It can be used only when the replacement product instance on the local node has the same set of identifiers as the one on the distribution node destined for removal. It takes the same command line elements as **upd addproduct**. See sections 24.1 *upd addproduct* and 24.3 *upd delproduct* for more information on those commands.

24.12.1 Command Syntax

For replacing with a product that is declared to a local database

An unwound product or a table file:

```
% upd reproduct <flavor_option> [<other_options>] <product> \
  <version>
```

A product tar file:

```
% upd reproduct <flavor_option> -T <archFilePath> \
  [<other_options>] <product> <version>
```



Note: One difference from the **upd addproduct** syntax: The flavor option is strictly required here; **upd reproduct** does not default to the current instance on the local system.

For replacing with a product that is not declared to a local database

An unwound product:

```
% upd repproduct [-P] <flavor_option> -r <prodRootDir> \
  -m <tableFileName> [-M <tableFileDir>] [<other_options>] \
  <product> <version>
```

A table file:

```
% upd repproduct [-P] <flavor_option> -m <tableFileName> \
  [-M <tableFileDir>] [<other_options>] <product> <version>
```

A product tar file:

```
% upd repproduct [-P] <flavor_option> -T <archFilePath> \
  -m <tableFileName> [-M <tableFileDir>] [<other_options>] \
  <product> <version>
```

Notes:

- The **-P** option is not strictly required, however it is recommended. In cases where the local database contains a product instance that also matches the specified options, **-P** ensures that **UPD** ignores the database and picks up the intended instance.
- If the product includes a table file, you must include the **-m** option specifying its name, as there is no default. You must also include **-M** if the table file is not in the current directory.

24.12.2 Options

See section 24.1.3 *All Valid Options* under 24.1 *upd addproduct*.

24.12.3 upd repproduct Examples

Add and then replace a locally-declared product using defaults

```
% upd addproduct foo v1_0 -2
```

UPD looks in \$PRODUCTS to find the product **foo** v1_0 for the **-2** flavor level of the local machine. If necessary, it makes a tar file, adds it to the default distribution node *fnkits*, and declares it there.

Let's say that after adding the product you discovered it contained the wrong set of executables. After fixing the problem on your local node, you need to replace the product instance on the distribution node. You could just run the command:

```
% upd repproduct foo v1_0 -2
```

24.13 upd update

The **upd update** command retrieves the table file, `ups` directory, or both, for a product instance, and by default for its product dependencies, from a distribution database for the purpose of updating the pre-existing one(s) on a local node.

Overwriting occurs only if the MODIFIED date in the version file that points to the table file on the distribution node is later than that in the corresponding local version file; file timestamps are not used for the comparison. The update will fail if there is no corresponding pre-existing component on the local node.



If you need to update *both* the `ups` directory and the table file for a product, do it at the same time (using the component list **table_file:ups_dir**). Otherwise, the MODIFIED date will not allow the second operation to succeed.

24.13.1 Command Syntax

```
% upd update [<options>] <componentList> <product> [<version>]
```

Components

table_file	the table file
ups_dir	the <code>ups</code> directory

24.13.2 Commonly Used Options

See section 24.13.3 *All Valid Options* for descriptions of each option.

-f <flavorList> Or one of **-0**, **-1**, **-2**, **-3**, **-4**, or **-H** (alone or together with one of **-0**, **-1**, **-2**, **-3**, **-4**)

-g <chainName> Or one of **-c**, **-d**, **-n**, **-o**, **-t**

-h <host>

-i

-I

-j

-q <qualifierList>

-R

-z <databaseList>

24.13.3 All Valid Options

- ? ("**-?**" for **cs**) Prints command description and option usage information to screen
- a Updates all instances that match the other options given on command line.
- c Finds product instance chained to "current" on local and distribution nodes
- d Finds product instance chained to "development" on local and distribution nodes
- f <flavorList> Described below under "The flavor options".
- g <chainName> Finds product instance chained to <chainName> on local and distribution nodes; can be a list
- h <host> Specifies product distribution host to which you are adding the product; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:
 - a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
 - a host and Webserver port number (e.g., *fred.sub.domain:8080*)
 - a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)
- H <flavorList> Described below under "The flavor options".
- i Ignores errors; if command is to be repeated for a list of flavors and errors are generated for one or if download of a component fails, operation continues to the next.
- I When updating a `ups` directory, unwinds the directory tar file on the local node after transmission (default is to unwind during transmission).
- j Ignores dependencies, operates just on top level product.
- n Finds product instance chained to "new" on local and distribution nodes.
- o Finds product instance chained to "old" on local and distribution nodes.
- O "<flags>" Sets the value of `$UPS_OPTIONS` to <flags>.
- q <qualifierList> Finds product instance on distribution node with the specified qualifiers (required and/or optional).

- R** Retrieves/operates on only the required dependencies.
- s** Lists what command would do; but does not execute the command
- t** Finds product instance chained to “test” on local and distribution nodes
- v(vvv)** Prints out extra debugging information.
- z <databaseList>** Specifies the local database(s)

The flavor options

Flavor may be specified using **-f**, using **-H** by itself or in combination with any of **-0**, **-1**, **-2**, **-3**, **-4**, or just using one of **-0**, **-1**, **-2**, **-3**, **-4**. These options are not valid with each other (except **-H** with a number option).



-f <flavorList> Finds product instance of specified flavor on local and distribution nodes. If multiple values specified, **upd update** is run once for each flavor. Avoid using **-f** when dependencies are involved, because it can cause flavor mismatches between main product and dependencies.

-H <flavorList> Specifies flavor and builds a flavor list for that family starting at the level specified. Multiple values can be specified, usually of different flavor families. If multiple values specified, **upd update** is run once for each value.

Can be used alone (without an accompanying number option). In this case, the best match instance of the flavor family is picked.

If used with any of **-0**, **-1**, **-2**, **-3**, **-4**, finds product instance with specified level of that flavor; e.g., **-2H IRIX+6.2** is equivalent to **-f IRIX+6**. If multiple values specified, **upd update** is run once for each flavor, according to the accompanying number option.

- 0** Specifies flavor as NULL; equivalent to **-f NULL**
- 1** Specifies flavor as OS value up to the generic OS; e.g., on a SunOS machine it is equivalent to **-f SunOS**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX.
- 2** Specifies flavor for product instance on local and distribution nodes as OS value up to the version; e.g., equivalent to **-f SunOS+5**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.

- 3 Specifies flavor for product instance on local and distribution nodes upto the release of the version; e.g., equivalent to **-f SunOS+5.6**; if given together with **-H IRIX+6.2**, flavor is then specified as IRIX+6.2.
- 4 Specifies flavor for product instance on local and distribution nodes upto the patch number of the release; e.g., equivalent to **-f SunOS+5.6.2**; if given together with **-H IRIX+6.2.1**, flavor is then specified as IRIX+6.2.1.

24.13.4 upd update Examples

Before running **upd update**, compare the MODIFIED dates for the product. To do so, run:

```
% ups list -K MODIFIED[:<other_keywords>] [<options>] <product>
[<version>]
```

on the local node and run **upd list** with similar options on the distribution node.

Update a table file

```
% upd update table_file xntp v3_4 -H SunOS:IRIX:OSF1:Linux
```

UPD updates the table file (the component **table_file** is specified) for the product **xntp v3_4** (and for all its dependencies) for the best match flavors to those listed. If **-f** were used in place of **-H** here, all the dependencies installed would be of the best match flavor to the local machine, and most or all product instances wouldn't work right. Output is provided indicating success or failure (in this case, success):

```
updcmd::updcmd_update - Updating xntp.
upderr::upderr_syslog - successful transfer
ftp://fnkits.fnal.gov//ftp/upsdb/xntp/v3_4SunOS.table ->
/tmp/mwmdb/xntp/v3_4.table
upderr::upderr_syslog - successful ups touch xntp v3_4 -f
SunOS -q "" -U ""
```

After performing the **upd update**, rerun the **ups list** command to verify the result.



Note: When updating several instances at a time, you can exclude a particular instance from being updated by running **ups touch** on it. See section 23.17 *ups touch* for more information.

24.14 upd verify

The **upd verify** command performs **ups verify** on a distribution database. **ups verify** checks the integrity of the local database files for the specified product(s), and lists any errors and inconsistencies that it finds. See section 23.20 *ups verify* for more information and examples.

24.14.1 Command Syntax

```
% upd verify -h [host] [<ups_verify_options>] [<product>] \
    [<version>]
```

24.14.2 Options

The **upd verify** command uses all the same options as **ups verify**, plus **-h** (described below). See section 23.20 *ups verify* for the list of options and their descriptions. The difference in the options' functionality is that they apply to the distribution node rather than to the local node.

-h <host> Specifies product distribution host; the default is *fnkits.fnal.gov*. Usually just the plain host name is required, however the **-h** option can always specify any of the following:

- a plain host name; (e.g., *fred.sub.domain*, or just *fred* if *sub.domain* is *fnal.gov*)
- a host and Webserver port number (e.g., *fred.sub.domain:8080*)
- a full URL to the `ups.cgi` cgi script (e.g., `http://fred.sub.domain:8080/cgi-bin/some/dir/ups.cgi`)

24.15 upp

UPP is a layer on top of **UPD** that can be used to facilitate the update of products on a local **UPS** node as new versions become available on a product distribution node. It has a single command, **upp**, which is controlled by a subscription file. The functions **upp** can be configured to perform on a local node include:

- notifying the client of new and updated products on a specified distribution node

- performing product installations and updates
- installing/updating product dependencies and resolving chains to maintain integrity of parent product
- deleting old product versions



We include this command here to document its syntax. The subscription file, which determines the functionality of the command, is described in Chapter 33: *The UPP Subscription File*.

24.15.1 Command Syntax

```
% upp [-v] <subscription_file_1> [<subscription_file_2>[, ...]]
```

24.15.2 All Valid Options

-v(vvv) Prints out extra debugging information.

24.15.3 upp Examples

```
% upp /path/to/upp.subscription
```

This command executes the subscription file `upp.launch`. No verbose output is requested (no **-v** option). This example is not terribly helpful for two reasons:

- it doesn't show how the subscription file controls **UPP** (beyond the scope of this section), and
- **UPP** is most often used in an automated way, e.g., using **cron**.

We refer you to Chapter 7: *Installing Products Using UPP* and section 11.5.3 *Using UPP to Remove a Product* for examples, and to Chapter 33: *The UPP Subscription File* for detailed information on subscription files.

A better example is to show how to execute the above command using **cron**. We'll add a **cron** job that runs a script we'll call `upp.launch`. This script first sets up **UPD** then runs the **upp** command as shown above:

```
#!/bin/sh
. /usr/local/etc/setups.sh
setup upd
upp /path/to/upp.subscription
```

A sample crontab entry to run the `upp.launch` script every night at midnight might look like:

```
0 0 * * /path/to/upp.launch
```